

Programmation réseau avec sockets en C sous Linux (1)

V. FELEA & A. HUGEAT & E. MERLET

Les exercices sur les sockets constituent une initiation à la programmation par sockets en C, pour la communication entre processus déployés sur des ordinateurs différents, reliés par une connexion réseau. Les tests peuvent être effectués dans un premier temps sur le même ordinateur local, dans de tels cas l'interface réseau locale (lo) étant utilisée (deux terminaux peuvent servir pour le lancement des deux applications communicantes). Dans un deuxième temps, grâce à une connexion ssh (voir TP2/section 2), le test sera effectué sur deux ordinateurs physiquement distincts, pour faire exploiter l'interface réseau Ethernet.

Ce premier exercice concerne la communication entre deux processus différents en mode connecté, utilisant le protocole de transport TCP.

Envoi d'une chaîne de taille fixe (sens unidirectionnel de communication)

Écrire une application qui envoie une chaîne de caractères, reçue par une autre application. La taille de la chaîne est fixe et connue par les deux applications (grâce à une constante). Dans cette application, c'est le serveur qui envoie le message et le client le reçoit et l'affiche. Le port du serveur est configuré par un argument en ligne de commande. Le nom de l'ordinateur serveur (ou son IP) et le numéro de port serveur sont des arguments en ligne de commande pour le client.

Indication. Dans la communication du message (la chaîne de caractères), ne pas oublier d'envoyer le marqueur de fin de chaîne de caractères qui est le caractère `'\0'`.

Nous considérons, sur l'ensemble des programmes à développer, la famille de sockets correspondant à une communication distante, pour le protocole internet IPv4.

Important. Les commandes système `ss/netstat` peuvent être utiles pour visualiser les connexions réseau.

- Q0 Récupérer l'archive (voir cours Réseaux Moodle) contenant les binaires du client et du serveur développés par nos soins. Lancer les deux binaires, dans deux terminaux différents - le serveur en premier, suivi du client - sur l'ordinateur local et analyser le résultat obtenu, en le confrontant aux attentes de l'exercice.
- Q1 Essayer de lancer une première instance, suivie d'une seconde (sans arrêter la première instance), du binaire serveur et commenter le résultat. Essayer de lancer le client en premier, suivi du serveur et commenter le résultat.
- Q2 Développer le client, en utilisant comme programme de test de côté serveur, le binaire fourni. Tester l'exécution de l'application sur l'ordinateur local.

Q3 Développer le serveur en utilisant comme programme de test de côté client le binaire fourni. Tester l'exécution de l'application sur l'ordinateur local.

Remarque. Des lancements successifs du serveur, sur un même port, peuvent occasionner des erreurs, car, malgré la fin de l'exécution du serveur, le système libère le port après un laps de temps. Dans ces cas, changer le numéro de port utilisé.

Important. Pour toutes les exécutions ultérieures de l'application client-serveur, prévoir un bilan : hôtes utilisés (local/distant) et résultat (fonctionnel ou non - en cas d'erreur, marquer l'erreur obtenue).

Q4 Tester les deux programmes écrits par vos soins sur l'ordinateur local.

Q5 Tester les deux programmes écrits par vos soins sur deux ordinateurs physiques distincts.

- lancement croisé avec un collègue qui utilise également un ordinateur de la salle de TP. Un d'entre vous lancera le serveur, l'autre le client.
- lancement sur deux ordinateurs différents du réseau 172.20.128/24 (réseau des ordinateurs des salles de TP du département informatique) de vos deux programmes développés. Pour cela, il faut que les programmes, ainsi que les binaires, soient dans le répertoire `/users/loginEnt`.

Remarque. L'utilisation du répertoire `/users/loginEnt` permet de retrouver les fichiers lors d'une connexion ultérieure. L'utilisation du répertoire `/home/AD/loginEnt` (celui par défaut d'une connexion ssh) ne synchronisera pas les données sur le compte réseau, et les fichiers seront perdus au redémarrage de l'ordinateur.

Q6 Lancer une instance de serveur sur l'ordinateur local et, sans l'arrêter, une autre instance de serveur sur un ordinateur distant. Commenter le comportement et le comparer avec la réponse à la question Q1. Quelle affirmation sur les sockets confirme-t-il ce test ?